CLAIMS

We claim:

1. In a computerized environment including a host computer having a host controller, a client driver, and a peripheral device for transmitting one or more data packets to the client driver through a protocol stack, a method for transferring data packets in a way that can reduce the overhead that can otherwise be associated with transferring data packets through a protocol stack, the method comprising the following:

at a client module, an act of initiating a data transfer request that will be sent to a peripheral device, the data transfer request including data request instructions;

an act of allocating a buffer that corresponds to the data transfer request;

an act of sending the data transfer request through a protocol stack, wherein the data transfer request is mapped to the allocated buffer, and wherein the request instructions are inserted into a schedule at the host controller;

an act of receiving requested data from the peripheral device, wherein the requested data are directed by the host controller to the allocated buffer;

upon receiving the requested data into the allocated buffer, an act of deactivating the data request instructions in the host controller schedule, wherein the deactivated data request instructions do not need to be removed from the host controller schedule; and

an act of sending a signal to the client module that the allocated buffer has been filled.

2. The method as recited in claim 1, wherein the peripheral device is one or more of a game controller, a personal device assistant, a wireless device, a keyboard, and a mouse.

3. The method as recited in claim 1, wherein the peripheral device communicates over one or more of a USB, Ethernet, Bluetooth, and HID communication protocol.

Docket No. 13768.434

4. The method as recited in claim 1, further comprising locking the allocated buffer into physical memory.

5. The method as recited in claim 4, further comprising removing the requested data from the allocated buffer, and sending a signal that the allocated buffer is available to be filled again.

6. The method as recited in claim 5, wherein the client module removes the requested data from the allocate buffer.

7. The method as recited in claim 5, wherein the client module is one or more of a client driver and an application program, and wherein the client module includes specific information about the peripheral device.

8. The method as recited in claim 5, further comprising sending a signal to the host controller that the allocated buffer is available to be filled again.

9. The method as recited in claim 8, wherein the signal to the host controller that the allocate buffer is available to be filled again further comprises a signal to activate the data request instructions in the host controller schedule, such that a new data request instructions do not need to be inserted into the host controller schedule for a new data request.

10. The method as recited in claim 1, wherein one or more of one or more of software layers in the protocol stack perform the acts of allocating the corresponding buffer, mapping the buffer to the data transfer request, and inserting the data request instructions in the host controller schedule.

11. The method as recited in claim 1, wherein the data transfer request includes one or more other data transfer requests packaged as a single data transfer request.

12. The method as recited in claim 11, wherein the data request instructions comprise one or more transfer descriptors that provide instructions corresponding to the data transfer request and each of the one or more other data transfer requests packaged as a single data transfer request.

13. The method as recited in claim 12, wherein the allocated buffer is a ring buffer comprising one or more buffer regions that correspond to the data transfer request and the one or more other data transfer requests.

Docket No. 13768.434

14. In a computerized environment including a host computer having a host controller, a client driver, and a peripheral device for transmitting one or more data packets to the client driver through a protocol stack, a method for transferring data packets in a way that can reduce the overhead that can otherwise be associated with transferring data packets through a protocol stack, the method comprising the following:

at a client module, an act of initiating a data transfer request that will be sent to a peripheral device, the data transfer request including data request instructions;

an act of allocating a buffer that corresponds to the data transfer request;

an act of sending the data transfer request through a protocol stack, wherein the data transfer request is mapped to the allocated buffer, and wherein the request instructions are inserted into a schedule at the host controller; and

a step for reducing the overhead associated with processing additional data transfer requests by recycling the allocated buffer, maintaining the data request instructions in the host controller schedule, and relaying information buffer availability information between the host controller and client device.

15. The method as recited in claim 14, wherein the step for reducing the overhead associated with processing additional data transfer requests comprises:

an act of receiving requested data from the peripheral device, wherein the requested data are directed by the host controller to the allocated buffer;

upon receiving the requested data into the allocated buffer, an act of deactivating the data request instructions in the host controller schedule, wherein the deactivated data request instructions do not need to be removed from the host controller schedule; and

an act of sending a signal to the client module that the allocated buffer has been filled.

Docket No. 13768.434

16. In a computerized environment including a host computer having a host controller, a client driver, and a peripheral device for transmitting one or more data packets to the client driver through a protocol stack, a computer program product having computer-executable instructions thereon for implementing a method for transferring data packets in a way that can reduce the overhead that can otherwise be associated with transferring data packets through a protocol stack, the method comprising the following:

> at a client module, an act of initiating a data transfer request that will be sent to a peripheral device, the data transfer request including data request instructions;

> an act of allocating a buffer that corresponds to the data transfer request;

> an act of sending the data transfer request through a protocol stack, wherein the data transfer request is mapped to the allocated buffer, and wherein the request instructions are inserted into a schedule at the host controller;

> an act of receiving requested data from the peripheral device, wherein the requested data are directed by the host controller to the allocated buffer;

> upon receiving the requested data into the allocated buffer, an act of deactivating the data request instructions in the host controller schedule, wherein the deactivated data request instructions do not need to be removed from the host controller schedule; and

> an act of sending a signal to the client module that the allocated buffer has been filled.

17. The computer program product as recited in claim 16, further comprising locking the allocated buffer into physical memory.

18. The computer program product as recited in claim 17, further comprising removing the requested data from the allocated buffer, and sending a signal that the allocated buffer is available to be filled again.

19. The computer program product as recited in claim 18, wherein the client module removes the requested data from the allocate buffer.

20. The computer program product as recited in claim 19, wherein the client module is one or more of a client driver and an application program, and wherein the client module includes specific information about the peripheral device.

21. The computer program product as recited in claim 17, further comprising sending a signal to the host controller that the allocated buffer is available to be filled again.

22. The computer program product as recited in claim 21, wherein the signal to the host controller that the allocate buffer is available to be filled again further comprises a signal to activate the data request instructions in the host controller schedule, such that a new data request instructions do not need to be inserted into the host controller schedule for a new data request.

23. The computer program product as recited in claim 16, wherein one or more of one or more of software layers in the protocol stack perform the acts of allocating the corresponding buffer, mapping the buffer to the data transfer request, and inserting the data request instructions in the host controller schedule.

24. The computer program product as recited in claim 23, wherein the data transfer request includes one or more other data transfer requests packaged as a single data transfer request.

25. The computer program product as recited in claim 24, wherein the data request instructions comprise one or more transfer descriptors that provide instructions corresponding to the data transfer request and each of the one or more other data transfer requests packaged as a single data transfer request.

26. The computer program product as recited in claim 25, wherein the allocated buffer is a ring buffer comprising one or more buffer regions that correspond to the data transfer request and the one or more other data transfer requests.

27. In a computerized environment including a host computer having a host controller, a client driver, and a peripheral device for transmitting one or more data packets to the client driver through a protocol stack, a computer program product having computer-executable instructions thereon for implementing a method for transferring data packets in a way that can reduce the overhead that can otherwise be associated with transferring data packets through a protocol stack, the method comprising the following:

at a client module, an act of initiating a data transfer request that will be sent to a peripheral device, the data transfer request including data request instructions;

an act of allocating a buffer that corresponds to the data transfer request;

an act of sending the data transfer request through a protocol stack, wherein the data transfer request is mapped to the allocated buffer, and wherein the request instructions are inserted into a schedule at the host controller; and

a step for reducing the overhead associated with processing additional data transfer requests by recycling the allocated buffer, maintaining the data request instructions in the host controller schedule, and relaying information buffer availability information between the host controller and client device.

28. The computer program products as recited in claim 27, wherein the step for reducing the overhead associated with processing additional data transfer requests comprises:

an act of receiving requested data from the peripheral device, wherein the requested data are directed by the host controller to the allocated buffer;

upon receiving the requested data into the allocated buffer, an act of deactivating the data request instructions in the host controller schedule, wherein the deactivated data request instructions do not need to be removed from the host controller schedule; and

an act of sending a signal to the client module that the allocated buffer has been filled.

Docket No. 13768.434